Hindawi
Complexity
Volume 2021, Article ID 5520324, 13 pages
https://doi.org/10.1155/2021/5520324

Research Article

Designing of a Simulation Tool for the Performance Analysis of Hybrid Data Center Networks (DCNs)

Muhib Ahmad,[1] **Farrukh Zeeshan Khan**,[1] **Zeshan Iqbal**,[1] **Muneer Ahmad**,[2] **Ihsan Ali**,[2] **Sultan S. Alshamrani**,[3] **Muhammad Talha**,[4] **and Muhammad Ahsan Raza**[5]

[1] *Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan*
[2] *Faculty of Computer Science & Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia*
[3] *Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[4] *Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia*
[5] *Department of Information Technology, Bahauddin Zakariya University, Multan 60000, Pakistan*

Correspondence should be addressed to Ihsan Ali; ihsanalichd@siswa.um.edu.my

Received 17 February 2021; Revised 13 March 2021; Accepted 24 March 2021; Published 10 April 2021

Academic Editor: M. Irfan Uddin

Data center (DC) technology changes the mode of computing. Traditional DCs consist of a single layer and only have Ethernet connections among switches. Those old-fashioned DCs cannot fulfill the high resource demand compared with today's DCs. The architectural design of the DCs is getting substantial importance and acting as the backbone of the network because of its essential feature of supporting and maintaining the rapidly increasing Internet-based applications which include search engines (e.g., Google and Yandex) and social networking applications (e.g., YouTube, Twitter, and Facebook). Every application has its parameters, like latency and blocking in the DC network. Every data center network (DCN) has its specialized architecture. It has a specific arrangement of layers and switches, which increase or decrease the DC network's efficiency. We develop a simulation tool that comprises two different DC architectures: basic tree architecture and c-Through architecture. Using this simulation, we analyze the traffic behavior and the performance of the simulated DCN. Our main purpose is to focus on mean waiting time, load, and blocking with respect to the traffic within the DCN.

1. Introduction

In this growing era of computing, thousands of applications are introduced; moreover, those applications had millions of users at a time. The people interacting with these applications belong to different domains. These millions of users interact with thousands of applications at a time daily. In this changing era of technology, the idea called "*Cloud Computing or Mobile Computing*" arose. Cloud computing or mobile computing introduces a completely new computing technology mode [1]. This new era of technology highly increases the number of applications that result in an increased number of users. Now, millions of people connect or interact and use them [2].

To assist the idea of this new era of technology, a mode of technology came named as datacenter (DC). The cost of the data center network is equal to the price of the old parallel computer network but with a huge difference in bandwidth, latency, reliability, security, modularity as well as power consumption. As the Internet world grows, the Internet and cloud computing-based applications have appeared. This speedy evolution of the internetwork and cloud computing and cloud computing-based applications demand maximum bandwidth, maximum throughput, and low latency [3]. Moreover, the interconnection also ensures the reliability of the network. The rapid emergence in Internet traffic due to the number of new emerging web applications has increased the demand for network bandwidth in the DCN. The traffic in the network employing using a particular application increases day-by-day exponentially. DC architectural design is getting substantial importance and acting as the backbone of the network because of its essential feature of supporting and maintaining the rapidly increasing internetwork-based applications which include large-scaled computations (e.g., indexing, bioinformatics, and data mining), web hosting, video hosting with its distribution (e.g., YouTube and NetFlix) as well as includes the huge number of social networking applications (e.g., Facebook and Twitter) [4]. The DCNs serve as a backbone in a new model of computing to meet the huge numbers of users' needs in a more efficient way.

From the last decades, the highest increase in the Internet traffic predominantly directed from egressing the number of web applications like social networking plus cloud computing bent the requirement of additional dynamic and energetic DCs. The servers will experience very low latency communicating if the DCs grow greater in size containing thousands of servers per rack. C set a major challenge related to DCs networking, constructing the required essential wish to build a new effective interconnection infrastructure with maximum bandwidth and minimum latency. As the size of the warehouse-scale (WS) data center increases, communication between the inter as well as the intradata center becomes more challenging in terms of inter as well as intradatacenter communication [5], power consumption and high bandwidth requirements, and small latency delay [3, 6].

2. DC Architecture

As technology grows, the different DC architectures emerged to support computing demands. Currently, the DCN architectures are based on a standard layered approach, as shown in Figure 1, which passes through several tests and has been improved over the past several years. The DC infrastructure's layered approach, sometimes also called the hierarchical internetworking model [2], includes the core layer, aggregation layer, and access layers.

The core layer is the uppermost layer in the design of DCN. This core layer is responsible for the fast transmission of data in the whole network that is why this core layer is recognized as the backbone of the DCN because all other layers depend on this core layer. This core layer is not responsible for routing traffic (packets) at the LAN. Moreover, the switches used in this layer do not deal with packet controlling. It has high-level duties in the network and takes most of the time designing the DCN [7].

This layer's main purpose is to provide minimum latency time in the delivery of packets. It consists of high-speed cables called fiber cables because of its high transmission rate and minimum reconfiguration time and high-end switches. The key term that the core layer majorly focuses on is efficiency. The essential features considered in designing the DC core layer switch are that the core layer's main and important factor is speed. The core layer provides a maximum data transfer rate through load sharing because multiple connections are used in the network through which the traffic travels. Multiple connections are used in the network to maximize the network's fault tolerance rate. In addition, the second factor is the core layer of DCN which deals with high-speed packet switching for all the ingress and egress packets in DCN. The core layer of DCN also offers fabric for high-speed packet switching among multiple aggregation modules in the DCN. It promises resilient fabric Layer 3 routed without any point of failure, as shown in Figure 1.

After the core layer in DC infrastructure, the aggregation layer is designed. It is the middle layer and acts as a medium to connect the core and access layers. The aggregation layer consists of Layer 3 switches and LAN-based routers. This layer makes sure that the packets are accurately transmitted among subnets and VLANs or in the network. This layer of the DCN is also known as the workgroup layer. Due to its huge number of tasks performed in DCN, it is also called the distribution layer or services layer. It provides vital functions, for example, service module integration, minimum spanning tree execution as default gateway redundancy [7]. Multi-tier traffic from server-to-server travels through the aggregation layer and utilizes multiple services, like firewall, load balancing, and optimization applications [8, 9].

In Figure 1, the block switches icons in the aggregation layer show the integrated service modules. The main aspect of the aggregation layer is routing. This means that the aggregation layer is responsible for route and regulates the packet transmission based on information of its origin and destination and to build a network border of communication. The layer provides an opportunity to create different protocol gateways for ingress and egress multiple network architectures. This layer helps and acts as the limit or edges for broadcast and multicast domains [10]. The Layer 3 switches or routers examine packets and then prioritize delivering the packets to the destination based on the policies and rules you set and applied.

The switches in the aggregation layer also have 10 GigE links with the switches, which are also used as ToR switch in the access layer. Moreover, it also has four pairs of 10 GigE links ($4 \times 10 = 40$ GigE) with other aggregation modules when using multiple aggregation modules in the data center network, as shown in Figure 1. The 40 GigE links to join the multiple aggregation modules give high bandwidth, avoid path blocking and quick convergence, and avoid the issues of overwhelming and arbitrarily broadcast.

In DCN, the bottommost and the last layer is access layer. It is the edge of the DCN where the switches called Top of Rack (ToR) switches are attached to the servers, as shown in Figure 1. The DC access layer offers physical connections to the servers and works in Layer 2 or Layer 3 modes. The primary oversubscription point is coming in the access layer because it aggregates the traffic of the server onto the 1 or 10 GigE channel uplinks in the DC aggregation layer. In the access layer, servers are physically connected to the network [7].

The modules of the server involve 1 rack unit (RU) servers, blade servers attached with integral switches, number of the blade, and clustered servers attached by pass-through cabling. Moreover, the access layer architecture also includes configured 1RU or 2RU switches, modular switches as well as integral blade server switches. Layer 2 and Layer 3-based architectures are done by switches used in this layer, by satisfying the server broadcast domains. The DC access

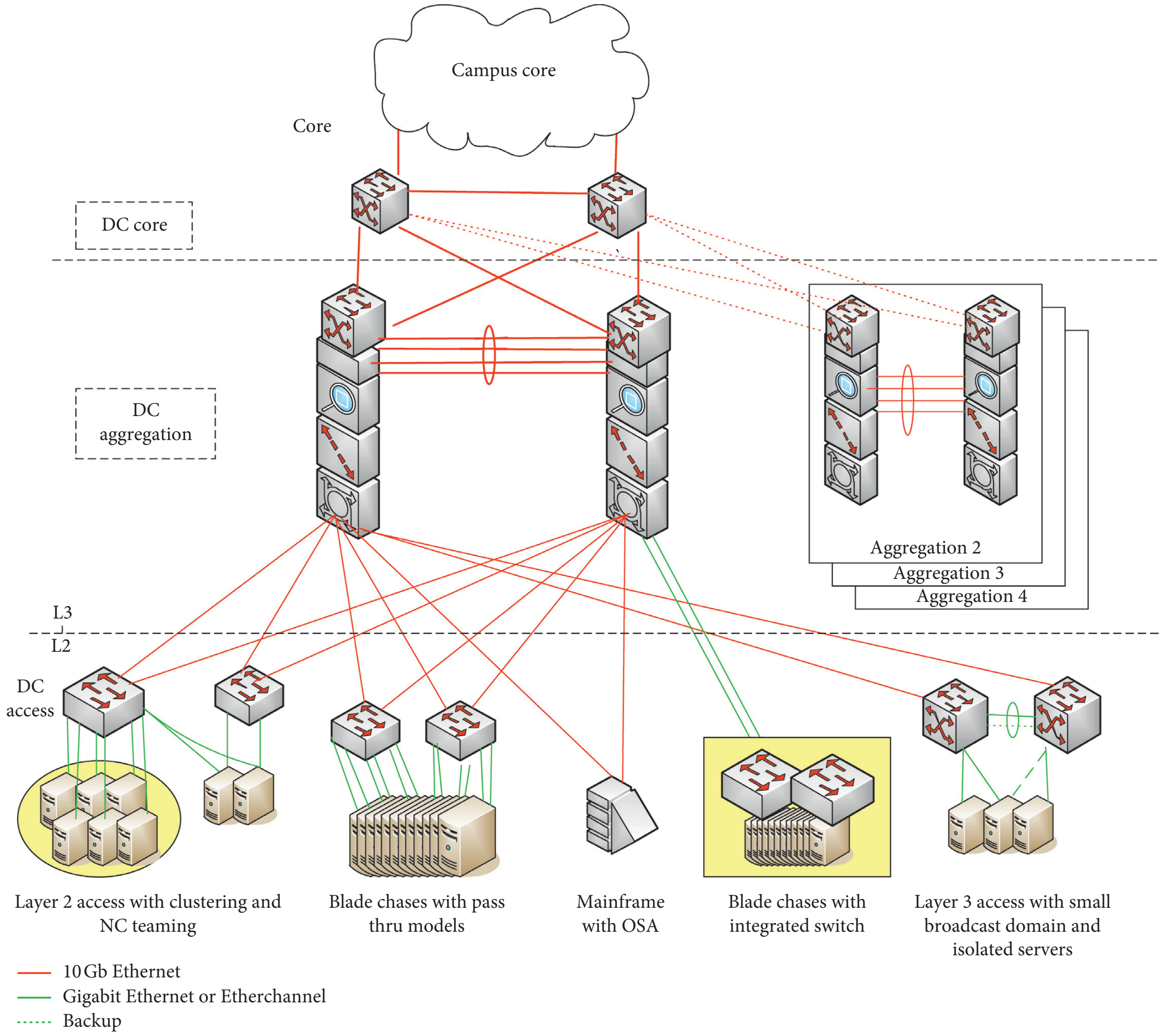

Figure 1: DCN layered architecture connected to different servers through ToR.

layer provides an opportunity to program a switch so that only particular systems on the network can access the related or joined LANs. Switch facilitates the nodes on the network by creating separate collision domains for every single connected node to enhance the performance of the network. By using load balancing, data can be moved from one network to another network.

In the access layer, ToR switches are usually used in every rack and have 1 GigE or 10 GigE slots connections to the network. These ToR switches connect to the number of servers that include the application servers, web servers, and data storage servers, as shown in Figure 1. This ToR architecture minimizes the cabling structure. These ToR switches are in greater numbers as compared with the switches used in the aggregation and the core layer. A single DC involves more than thousands of servers, arranged in a rack of 20–40 servers each [11, 12]. Servers within each rack are attached to the ToR switch, as shown in Figure 2, which are then further connected to the layers of clusters of switches in DC. The ToR switches connect to the aggregation layer switches through various high-speed connections of 10 GigE that are clustered to one another in a port channel. The packet travels through the core and aggregation layer. Based on the information in the packet, ToR examines and analyses the packets and fetches the data from the required server. All the information is encapsulated in the packet [7, 13]. Generally, less than 25% of average network peak load is held by DCNs, and large number of links in DCN stay idle for almost 70% of the time [14].

The important issue in DC traffic management is load balancing intended as reducing packet delay in internetwork and intranetwork; moreover, it is also able to redirect the flows by assessing energy-efficient targets. The primary factor of energy consumption in DCs is servers, and huge number of running servers results in higher power consumption. The longer the traffic stays in the DCN, it not only affects the energy consumption but also affect packets delay. Hence, the optimization of DCN servers utilization results in

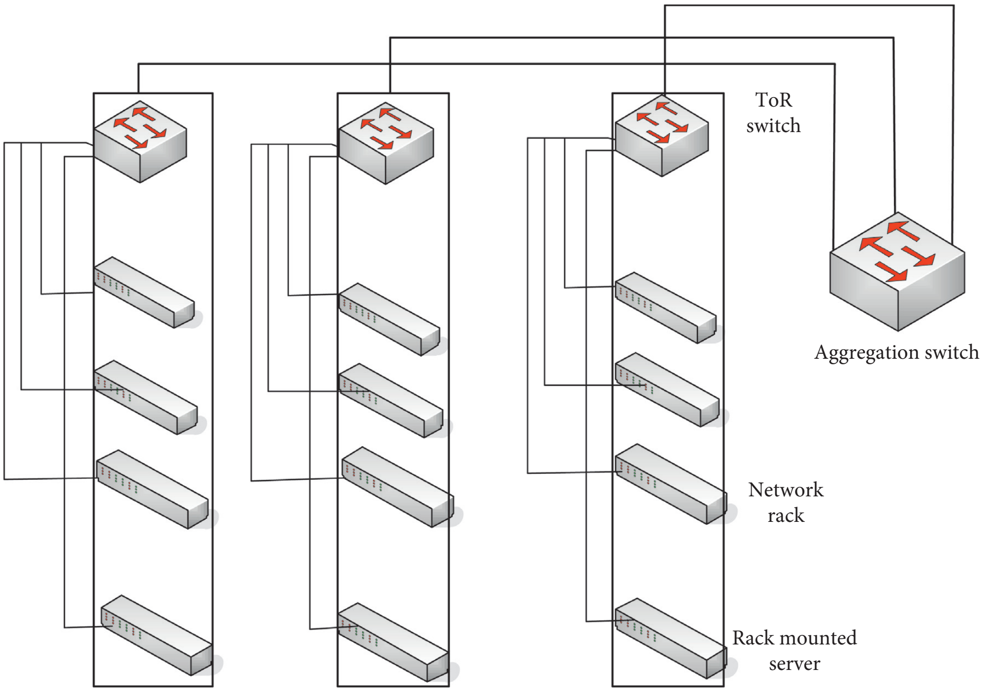

Figure 2: ToR switch connection with each server in the rack.

more efficient energy savings. Currently, a new hot research area in DCs is virtual machines (VMs) placement. Through appropriately placing VMs w.r.t the network and applications resource requirements, the network manager is able to save the superfluous resources for the other services and make the DCN much energy-efficient [15]. In this paper, we are not considering the whole network and placing the VMs. We are just modeling the traffic within the DC network [8, 16]. VMs placement and task distribution on different VMs are not considered because we only consider the traffic within the DCN (inter and intra DC network traffic).

3. Optical and Hybrid Data Center Architecture

3.1. OSA. The first design that uses optical switching technology in DCN is named OSA, as shown in Figure 3. OSA DC architecture is a "pure" optical switching network, which means that this architecture left the idea of switches based on electrical core and uses the idea of optical switches to build the switching cores [17].

While the ToR switches are electrical so far, so to transform and operate into optical-based and electrical-based signals among different servers and switching cores. In OSA architecture, the switching core has several connections on each ToR switch. The arrangement of connections is considerably flexible depending on the traffic demand.

Meanwhile, the architecture remained unable to provide the direct optical links between each rack pair according to the traffic demands, the design built by the controlling system creates a connected graph topology, and the ToR switches are liable to deliver the traffic among other ToR switches. Therefore, in OSA architecture, several optical links are provided to each rack [4].

3.2. c-Through. Wang et al. proposed the first hybrid electrical-optical network model data center network named c-Through. It is a three-level multirooted hybrid electrical and optical network model of DCN as shown in Figure 4, as an advancement to the current DCNs. c-Through is also known as HyPaC (hybrid packet and circuit) DCN. In c-Through architecture, ToR switches are linked to both networks made from electrical packet-based networks (Ethernet) and optical circuit-based networks [18].

This shows that it has two major parts: first, a tree-like electrical network that keeps the connectivity among every pair of switches of ToR and the second is a reconfigurable optical-circuit network that provides maximum bandwidth among various rack of servers. The optical switch is configured so that each pair of the rack, with a high bandwidth request, is attached with this optical switch. Due to the high cost of optical networks and promising maximum bandwidth between the optical links, there is no need to set the optical links to every single pair of racks, increasing the cost of the network [19]. In c-Through network, the optical part's configuration depends upon the traffic among the different types of racks of servers. c-Through architecture evaluates the rack-to-rack traffic demand by detecting the socket buffer's habitation. Each rack has a single optical link, so the whole architecture is reconfigurable at any time.

A traffic monitoring system is required that is positioned in hosts; thus, it calculates the bandwidth demand with the other hosts in the network. The optical configuration manager accumulates these measurements plus understands the optical switch configuration that depends on the traffic requirements. The nature of traffic requirements and the attached links are all expressed as maximum weight-perfect matching problems. That is why, in the c-Through network,

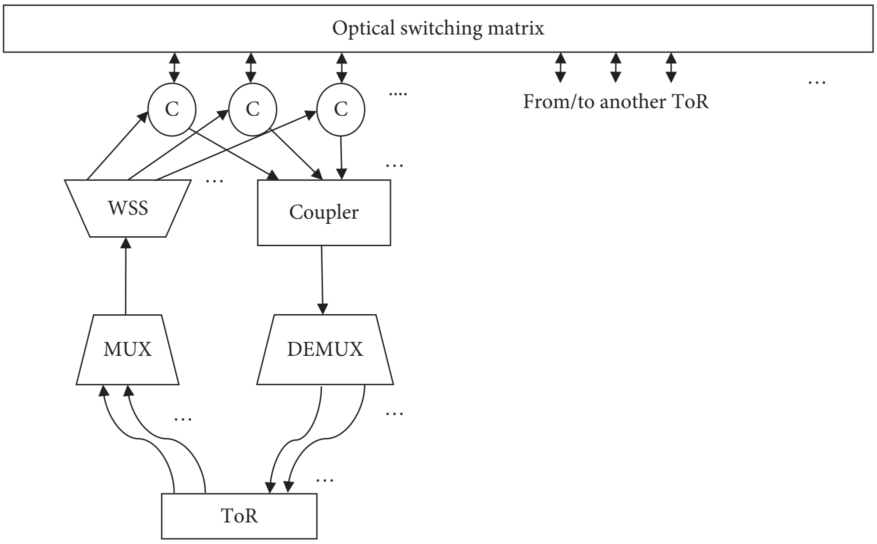

Figure 3: OSA architecture.

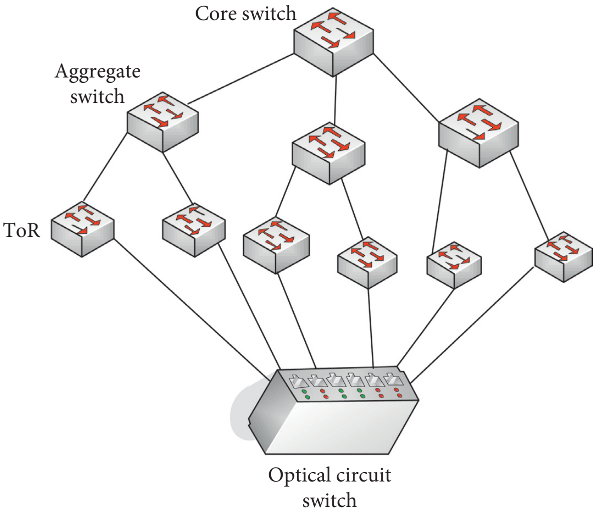

Figure 4: c-Through architecture.

the Edmonds algorithm is used as the key of the maximum weight-perfect matching problem [4]. After the optical switches' configuration, the optical manager notifies the ToR switches to route the packet based on information. Traffic in ToR switches is demultiplexed through virtual local area network (VLAN)-based routing [20].

Two different types of VLAN-based routing are used because of the electrical-optical network model: one is for packet network-based routing and the second for the optical circuit (OC) network-based routing [18].

3.3. Helios. In 2010, Farrington et al., from UCSD proposed an electrical-optical switched network model for modular DCN as mentioned in [4], which is known as Helios. Helios is the other hybrid network based on the electrical-optical network model just like the c-Through design architecture; however, the main difference in that architecture is that it is based on wavelength division multiplexing (WDM) links [4], as shown in Figure 5.

It is a two-level multirooted hybrid electrical and optical network model of DCN. This hybrid architecture consists of core switches and ToR switches known as Pod switches. The Pod or ToR switches are electrical packet-based switches. In contrast, on the other hand, the core switches are either the electrical packet-based switches or maybe the OC (optical circuit)-based switches [19, 21]. Thus, the electrical packet-based switches are all involved in all ToR switches communication.

In contrast, OC-based switches are employed for varying large bandwidth communication among ToR switches. Helios architecture uses electrical packet-based switching to allocate the network's erupt type of traffic flow. In contrast, OC-based switched network facilitates slow-changing traffic with high bandwidth. Each pod or ToR switch consists of colorless optical transceivers and WDM-based optical transceivers. The uncolored optical transceivers link the pod switches and the core electrical packet-based switches. Contrary to this, optical multiplexer multiplexes the WDM optical transceivers that form the super links attached to the OC switches.

The main disadvantage of this suggested approach is that it consists of micro-electro-mechanical systems switches (MEMS); therefore, the circuit switch wants various times (in milliseconds) in case of any reconfiguration (the reconfiguration time of Glimmerglass switch was 25 ms). Hence, this approach is perfect for those applications where connections among some nodes end at more than a few seconds to facilitate the reconfiguration overhead [19]. Among all the uplinks, half of the uplinks are attached to electrical switches and the remaining half uplinks are attached to the optical switches through the optical multiplexer. These super links are capable of transmitting $w \times 10$ Gigabit per second ($w =$ total number of wavelengths; starting from 1 to 32) [19].

Helios architecture's major advantage is that its architecture is built on easily accessible optical transceivers and modules commonly used in optical telecommunication

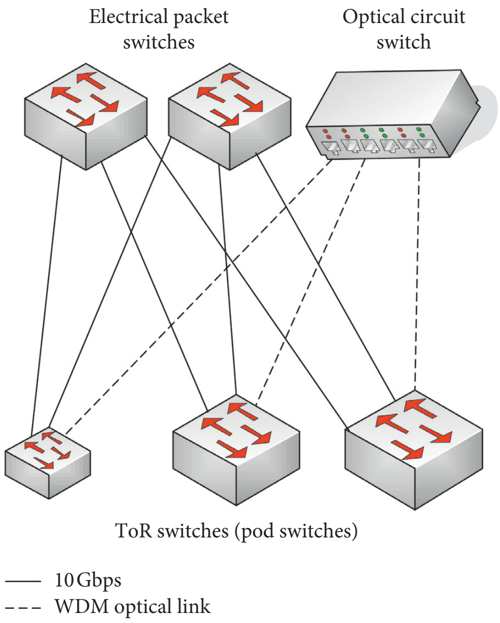

Figure 5: Helios architecture.

networks [19]. The OC switch consists of Glimmerglass-based switches, and ToR switches use the WDM SFP + optical transceivers. These transceivers of WDM are either dense WDM modules that have great cost and support greater ranges of wavelengths or coarse WDM that have less cost and support fewer wavelengths and use a broad spacing of channel (20 nm channels throughout 1270 nm to 1630 nm C-band). Coarse WDM (CWDM) laser does not demand costly temperature stabilization. Glimmerglass crossbar OC switch is used to calculate performances that are easily available and support equal to 64 ports. The authors in [22] introduced a new hybrid DCN model named TIO that rejects the previous optical/electrical switching scheme between racks and introduces a novel technique called visible light communication (VLC) between racks. It combined both wireless-based VLC Jellyfish architecture and wired-based EPS Fat Tree.

4. Traffic Modeling

Stimulations are performed to determine the traffic behavior and pattern between aggregate switches and ToR switches and the network connectivity connected through links. After performing several simulations, we get the values (readings) of packet arrival rate in aggregate switches. And measure the ToR switches performance on specific parameters, i.e., packet size, link capacity, and packet generate rate. The ingress packets are sent from the aggregate switch to ToR switch. After fulfilling its request from the server at a specific time, they came back to aggregate switch through ToR switch. On behalf of the DC parameters, we analyze the traffic behavior on the aggregate switches and the ToR switches as follows: mean waiting time-up, mean waiting for time-down, blocking-up, and blocking-down [20]. In DCN, burstiness of traffic has two different aspects: either the traffic burstiness per flow could be extremely bursty or the flow arrival itself could be bursty too. The intensity of burstiness of traffic may change corresponding to the point where the traffic is measured, that is, at end-point ports or at ToR switches. Because a variety of applications share the DC infrastructure, a combination of numerous flow types and sizes are produced. Flows may or may not have deadline limitations and may be short latency sensitive or large throughput oriented. Like some particular applications, flow sizes are unidentified. The brief summary of DC traffic control challenges is in [23, 24].

4.1. Load. The packets that arrived from the aggregate switch to ToR switch are through a link of 10 GbE. This arrival of packets between the switches through a link decides the load on the network's network and performance. In our basic tree model, the alternative path between ToR and ToR is through the aggregate switch. On the other hand, the alternative path in c-Through in through OCS manager is an optical link. So, we calculate the load ρ by applying the formula:

$$\rho = \frac{E\left[L_p\right]}{C} \times \frac{N_a}{t_{tr}}. \tag{1}$$

Here, $E[L_p]$ is the average packet size, C is the link capacity, N_a is the total number of packets arrived, and t_{tr} is the total transmission time. Contrary to this, load on the downward traffic is denoted as ρ_d and can be calculated as follows:

$$\rho_d = \frac{E\left[L_{pd}\right]}{C} \times \frac{N_{ad}}{t_{tr}}. \tag{2}$$

Here, $E[L_{pd}]$ is the average packet size-down, N_{ad} is the total number of downward packets arrived, and t_{tr} is the total transmission time.

While on the other hand, the packets that are considered as downward traffic are calculated as load-up ρ_u:

$$\rho_d = \frac{E\left[L_{pu}\right]}{C} \times \frac{N_{au}}{t_{tr}}. \tag{3}$$

Here, $E[L_{pu}]$ is the upward average packet and N_{au} has the total number of upward packets arrived.

4.2. Mean Waiting Time. On behalf of the load and average packet size $E[L_p]$, we calculate the mean waiting time of both types of ingress and egress traffic.

4.3. Blocking. Blocking is defined as the number of packets dropped N_d on a particular link during the transmission of packets. So, we calculate the blocking probability P_b at a particular link by the following formula:

$$P_b = \frac{N_d}{N_a}. \tag{4}$$

Here, N_a is the total number of packets arrived and N_d is the number of packets dropped on a particular link. The number of packets dropped on the upward traffic N_{du} is calculated as follows:

$$P_{bu} = \frac{P_{Du}}{N_{au}}. \tag{5}$$

Here, N_{du} is the total number of packets dropped on upward traffic and N_{au} represents the rate of arrival total number of upward packets. While on the other hand, the packets dropped on the downward traffic N_{dd} is calculated as follows:

$$P_{bd} = \frac{N_{dd}}{N_{ad}}. \tag{6}$$

Here, N_{dd} is the total number of packets dropped on downward traffic and N_{ad} represents the rate of arrival of the total number of downward packets.

5. Simulation Structure

There are several open-source simulators to analyze the performance and characteristics of the DC and also provide extra features of cloud computing and VM migration. Some open-source simulators are discussed here that are freely downloadable. Table 1 shows the comparison between open-source cloud simulators.

5.1. CloudSim. The idea of cloud computing introduces a completely new mode of computing technology with promising assurance to provide sustainable, secure, fault-tolerant, and reliable services, which are introduced as software and infrastructure as a service (SaaS and IaaS) [25]. To ensure these factors, a Java-based simulation model is developed at CLOUDS Laboratory, University of Melbourne, Australia, called CloudSim. It is a well-known simulator for analyzing large-scale cloud DC parameters. Researchers used this simulation framework to analyze the cloud infrastructure and explore application services' performance in a controlled environment. It provides basic classes for describing users, VMs, computational resources, DCs, and many other user and management level controls of infrastructure. The key drawback of the CloudSim simulator is the lack of graphical user interface (GUI) [26].

5.2. GreenCloud. GreenCloud is another open-source simulator, and it is an extension of network simulator NS2. This simulator is based on CloudSim. This open-source simulator allows the researchers and users to interact, watch, and control the DCs and clouds' performance. The green cloud computing revolution developed this simulation tool and gave an option to examine the energy efficiency of the cloud DCs. It also provides information about the DC equipment about their energy consumption [26]. This simulation framework distinguishes the energy consumption parts related to the infrastructure and computing energy of DC [25]. GreenCloud simulator is also used to build up new scheduling workload scenarios, allocating resources and optimizing network infrastructure and communication protocols. It majorly focuses on energy consumption issues of the DCN. The major limitation of GreenCloud is that it works only with small DCs.

5.3. EMUSIM. This tool has an integrated architecture of emulation and simulation. This simulator framework is built on AEF (automated emulation framework) technology for emulation and used CloudSim architecture for simulation. It contains both emulation and simulation techniques in a single package. This tool is used to design an environment near-real computing patterns and resources. EMUSIM simulator does not need lower-level details like the location of VM and total VM per host in a given time. The drawback of the EMUSIM is that it is not scalable due to many hardware restrictions or issues in producing real large networks.

5.4. iCanCloud. iCanCloud simulator is mainly used to determine the connection between the cost and the throughput of a given set of applications on a specific hardware configuration. This simulator is built on SIMCAN simulator [27], a simulation tool used to investigate the high-performance I/O architectures. This simulator gives information about the specific parameters of DCN [26].

5.5. IBKSim. For the simulation modeling of two different DC architectures, a simulator is designed to analyze the DCN parameters bandwidth, latency, and throughput. IBNSim is a network simulator developed in 2004 by using an object-oriented design model. With time, various new protocols were added to this simulator for advanced research purposes in the network field. IBKSim is the current simulator edition for research purposes and detailed in [28]. This simulator can simulate small queuing models and judge the performance of the single and big network nodes. It is a flexible simulator and uses the object-oriented design methodology in C++ development.

6. Simulation Setup

After developing our simulator, two different layered-based DC architectures were built: first is Ethernet-based called basic tree DC architecture and the other is hybrid architecture called c-Through architecture. Both architectures' design is almost the same as one core switch connected with three aggregate switches through a 10 GbE link. Then, each aggregate switch is attached with two ToR switches with 10 GbE links, and each ToR is connected with exactly one rack. The difference in both architectures is that, in basic tree architecture, the link between the aggregate switches is 40 GbE, which is not present in c-Through architecture. And the other difference is that, in c-through architecture, ToRs are connected with an optical circuit switch (OCS) with a 10 Gbps link that makes it hybrid.

6.1. Simulation in Basic Tree. When the simulation starts, the source generates packets, and these packets are transmitted to the core switch. The shortest path algorithm is used for routing in the core switch to minimize the network load. After receiving the packets from the source, the core node transmits packets to the intended aggregate switch based on

Table 1: Comparison between open-source cloud simulators.

Parameters	CloudSim	GreenCloud	EMUSIM	iCanCloud	IBKSim
GUI support	N/A	Limited support	Limited support	Full GUI support	Limited support
Language	Java	C++, Otcl	Java	OMNet, MPL, C++	C++, XML
Platform	GridSim	NS2	AEF, CloudSim	SIMCAN	N/A
Power efficiency	Limited	Yes	Yes	Limited	Limited
Simulation time	Seconds	Tens of minutes	Tens of minutes	Tens of minutes	Depend upon network size
Multi DC comparisons	No	No	No	No	Available
Support for parallel experiments	No	No	No	Available	Available

the routing table defined in the core switch. The flow chart shows how traffic travels in the DC network simulator that we developed, as shown in Figure 6.

Packets are arranged in the queues when arrived in the aggregate switch. Aggregate switch takes the routing information from the core switch and then forwarded the packets to the ToR switch according to the queue. The ToR switch checks the packet internal information that further sent to the rack if the packet looking for is found in the server if not, then the packet travels back from ToR to the aggregate switch. This aggregate switch sends to another aggregate switch for the intended server, as shown in Figure 7. The maximum traffic in this type of DC architecture in between the TORs and the aggregate switches is compared with the hybrid architecture. Simulation ends until either the input simulation time is completed or the number of times the specific event has occurred.

6.2. Simulation in c-Through. When the simulation starts, the source generates packets, and these packets are transmitted to the core switch. The shortest path algorithm is used for routing in the core switch to minimize the network load. The flow chart shows how traffic travels in the DC network simulator that we developed, as shown in Figure 8:

After receiving the packets from the source, the core node transmits packets to the intended aggregate switch based on the routing table defined in the core switch. Packets are arranged in the queues when arrived at the aggregate switch. Aggregate switch takes the routing information from the core switch and then forwarded the packets to the ToR switch according to the queue. The ToR switch checks the packet's internal information which further forwarded to the rack if the information which the packet looking for is found in the server if not, then the packet EOE (electrical-optical-electrical) conversion occurs and the electrical packet is converted into optical and forwarded to optical circuit switch. After that, the optical packet is again converted from optical to electrical and transmitted to another alternative ToR, as shown in Figure 9. The simulation ends until either the input simulation time completes or the specific event occurs.

6.3. Simulation Parameters. At the start of the simulation, the source generates the 2000 packets, increasing with the multiple of 2 per simulation and sending it to the core switch. The core switch uniformly distributes the packets and transmits them to the aggregate switch through a link of 10 GbE. Packets travel from aggregate switch to ToR switch through a link of 10 GbE. Racks are connected to TORs through a link of 10 GbE. In c-Through DC architecture, an optical link of 10 Gbps is used between two TORs through OCS while on the other hand, in basic tree architecture, the link between the two aggregate switches is 40 GbE. The mean packet size in our simulation is 0.001953125, and at each switch or node, the maximum packet limit of the queue is 1000.

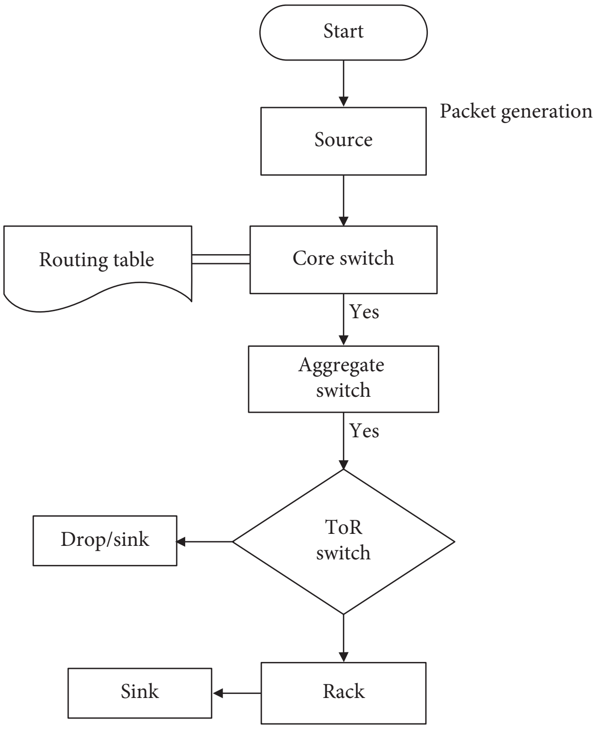

Figure 6: Flow chart of basic tree DCN simulation.

7. Results and Discussion

In this graph, mean waiting time-down means the flow of traffic moving downward from the core to aggregate and from ToR to rack or OCS whereas load-down represents a load of downward traffic on a link. The graph shows the behavior of downward moving traffic against the load.

The graph shown in Figure 10 represents the behavior of downward traffic on the aggregate layer. In the graph, the dotted lines of the tree-based architecture show that the mean waiting time-down and the load-down increases on an aggregate layer as compared with the c-Through

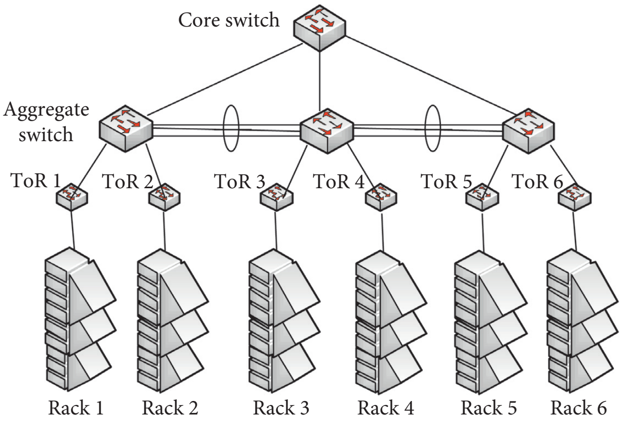

FIGURE 7: Simulated basic tree DC architecture.

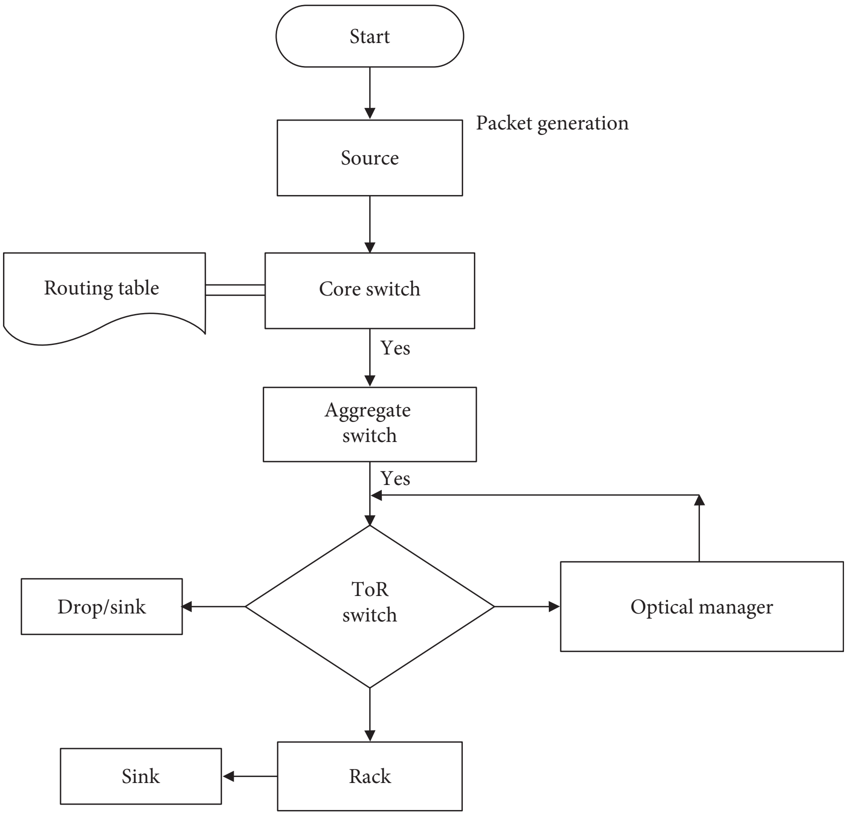

FIGURE 8: Flow chart of hybrid DCN simulation.

hybrid architecture because, in tree architecture, the packets arrive at the aggregate switch from the core and other aggregate switches. Packets are arranged in queues. The more the packets come, the more the previous packets have to wait in the queue, which increases the mean waiting time of the downward traffic. Mean waiting time-down increases more rapidly in tree architecture than hybrid architecture. Hybrid architecture increases less gradually and shows more consistent behavior.

While on the other hand, the graph in Figure 11 shows the behavior of downward traffic in ToR layer. The dotted lines of the tree architecture show that the mean waiting for time-down and the load-down increases with a much higher rate on ToR layer as compared with the c-Through hybrid architecture and show inconsistent manners. This is because transmission of packets from the aggregate switch despite treating the packets that are already in the ToR switch's queue increases the load on the ToR switch. Moreover, in tree architecture, packet communication between ToRs is through aggregate switches that also increase the load on ToR switch while on the other hand, c-Through has OCS as its alternative route among the ToRs communication. That is also a factor of inconsistent behavior of tree architecture.

The graph in Figures 12 and 13 shows the behavior of upward traffic on the aggregate and ToR layer. In tree architecture, the mean waiting time-up and the load-up in

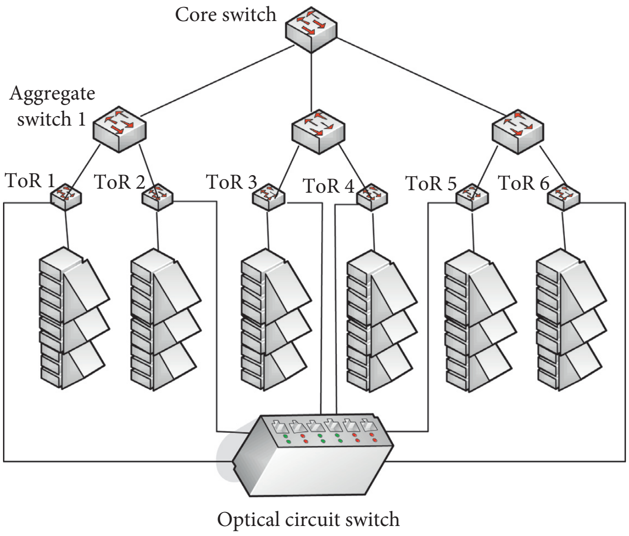

Figure 9: Simulated hybrid DC architecturee.

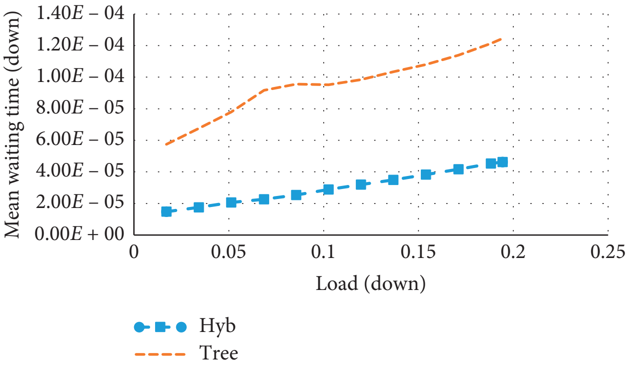

Figure 10: Mean waiting time-down of aggregate layer.

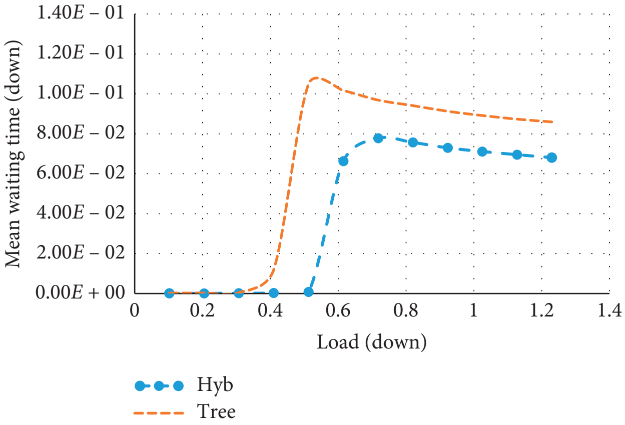

Figure 11: Mean waiting time-down of ToR layer.

both aggregate and ToR layers increase in an inappropriate manner. At the start, it first remains constant in both switches and then gradually increases. This gradual increase in aggregate is due to the reason the traffic between core to aggregate is much less than aggregate to ToR while in the case of hybrid, treating packet rate is greater than tree architecture and traffic between core to aggregate is greater than tree architecture. In Figure 13, the graph shows an increase in a discrepant manner and shows variance behavior. In tree architecture, packet communication between ToRs is through aggregate switches. So, the communication among ToR through aggregate switches increases the mean waiting time-up and load-up whereas c-Through architecture contains OCS as its alternative route among the ToRs communication so the mean waiting time-up of the hybrid on the graph is a straight line. That is also a factor of inconsistent behavior of tree architecture.

In Figures 14 and 15, the graph shows the blocking-down and the load-down in ToR layer, and in ToR layer, hybrid increases in a more consistent manner than tree architecture while in the aggregate layer both layers show the same behavior and pattern.

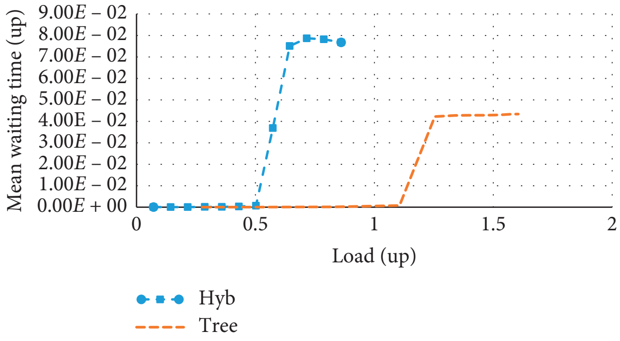

Figure 12: Mean waiting time-up of aggregate layer.

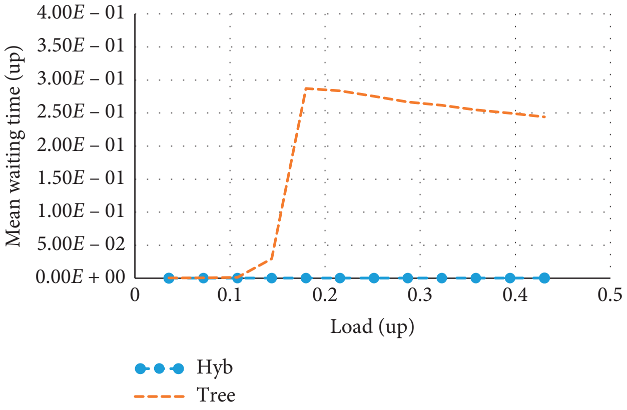

Figure 13: Mean waiting time-up of ToR layer.

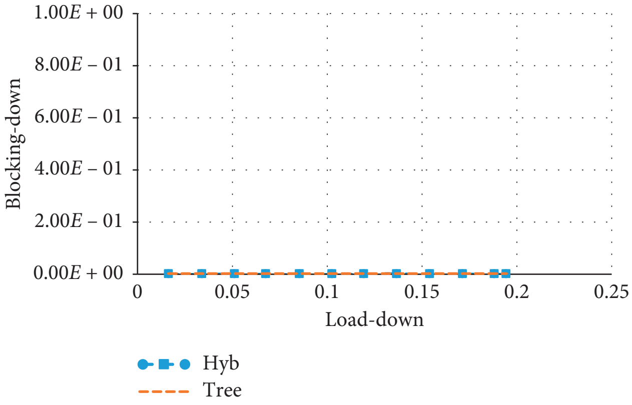

Figure 14: Blocking-down of aggregate layer.

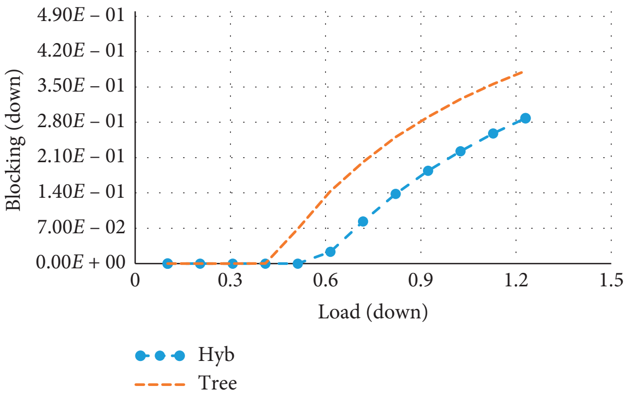

Figure 15: Blocking-down of ToR layer.

8. Conclusion

For this intuitive comparison, we first develop the tool that specifically measures the DC architecture parameters. We explore each switch component in the layered architecture and its characteristics. Furthermore, we analyze the load and blocking and mean waiting for time behavior of aggregate and ToR switches in both architectures through their traffic behavior in both switches. In the future, we can perform modeling on other network parameters through this tool. We can compare traffic and load parameters c-Through architecture with other fixed and flexible topologies of the DCN. The development of this research tool will help in the future for further comparisons with other DC architectures.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the Deanship of Scientific Research and King Saud University for funding this research work. The authors are also grateful to the Taif University Researchers Supporting Project number (TURSP-2020/215), Taif University, Taif, Saudi Arabia. This work was also supported by the Faculty of Computer Science and Information Technology, University of Malaya under Postgraduate Research Grant (PG035-2016A).

References

[1] S. Basu, C. McArdle, and L. P. Barry, "Scalable OCS-based intra/inter data center network with optical ToR switches," in *Proceedings of the 2016 18th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Warsaw, Poland, July 2016.

[2] B. Wang, Z. Qi, R. Ma, H. Guan, and A. V. Vasilakos, "A survey on data center networking for cloud computing," *Computer Networks*, vol. 91, pp. 528–547, 2015.

[3] L. Paraschis, "Advancements in data-center networking, and the importance of optical interconnections," in *Proceedings of the 39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, pp. 1–3, London, UK, September 2013.

[4] N. Farrington, G. Porter, S. Radhakrishnan et al., "Helios: a hybrid electrical/optical switch architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 339–350, 2010.

[5] S. Kumar, G. Papen, K. Schmidtke, and C. Xie, "Intra-data center interconnects, networking, and architectures," in *Optical Fiber Telecommunications VII*, pp. 627–672, Elsevier, Amsterdam, Netherlands, 2020.

[6] N. Panahi, D. Careglio, and J. S. Pareta, "A hybrid packet/circuit optical transport architecture for DCN," in *Proceedings of the 2019 21st International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Angerrs, France, July 2019.

[7] A. Headquarters, "Cisco data center infrastructure 2.5 design guide," *Cisco Validated Design*, vol. I, 2007.

[8] J. Zhang, F. R. Yu, S. Wang, T. Huang, Z. Liu, and Y. Liu, "Load balancing in data center networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2324–2352, 2018.

[9] M. Shafiee and J. Ghaderi, "A simple congestion-aware algorithm for load balancing in datacenter networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3670–3682, 2017.

[10] D. Li, Y. Li, J. Wu, S. Su, and J. Yu, "ESM: efficient and scalable data center multicast routing," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 944–955, 2011.

[11] C. F. Lam, "Optical network technologies for datacenter networks," in *Proceedings of the National Fiber Optic Engineers Conference*, Los Angeles, CA, USA, March 2010.

[12] C. Kachris, K. Kanonakis, and I. Tomkos, "Optical interconnection networks in data centers: recent trends and future challenges," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 39–45, 2013.

[13] Cisco, *Classic Network Design Using Cisco Nexus 9000 Series Switches*, Cisco, San Francisco, CA, USA, 2015.

[14] K. Bilal, S. U. R. Malik, O. Khalid et al., "A taxonomy and survey on green data center networks," *Future Generation Computer Systems*, vol. 36, pp. 189–208, 2014.

[15] D. Belabed, S. Secci, G. Pujolle, and D. Medhi, "Striking a balance between traffic engineering and energy efficiency in virtual machine placement," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 202–216, 2015.

[16] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, pp. 750–757, Honolulu, HI, USA, June 2012.

[17] K. Chen, A. Singla, A. Singh et al., "OSA: an optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Transactions on Networking*, vol. 22, pp. 498–511, 2013.

[18] G. Wang, D. G. Andersen, M. Kaminsky et al., "c-through: part-time optics in data centers," in *Proceedings of the ACM SIGCOMM 2010 Conference*, pp. 327–338, New Delhi, India, September 2010.

[19] C. Kachris and I. Tomkos, “A survey on optical interconnects for data centers,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1021–1036, 2012.

[20] D. Wu, *Designing Hybrid Data Center Networks for High Performance and Fault Tolerance*, Rice University, Houston, TX, USA, 2020.

[21] Y. Liu, J. K. Muppala, M. Veeraraghavan, D. Lin, and M. Hamdi, *Data Center Networks: Topologies, Architectures and Fault-Tolerance Characteristics*, Springer Science & Business Media, Berlin, Germany, 2013.

[22] Y. Qin, D. Guo, G. Tang, and B. Ren, “TIO: a VLC-enabled hybrid data center network architecture,” *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 484–496, 2019.

[23] M. Noormohammadpour and C. S. Raghavendra, “Datacenter traffic control: understanding techniques and tradeoffs,” *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 1492–1525, 2017.

[24] A. Saeed, N. Dukkipati, V. Valancius, V. The Lam, C. Contavalli, and A. Vahdat, “Carousel: scalable traffic shaping at end hosts,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 404–417, Los Angeles, CA, USA, August 2017.

[25] U. Sinha and M. Shekhar, “Comparison of various cloud simulation tools available in cloud computing,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, pp. 171–176, 2015.

[26] M. S. Mandeep and S. Bhatia, “A critical review & analysis of cloud computing simulators,” *IJLTET*, pp. 29–36, 2016.

[27] R. Malhotra and P. Jain, “Study and comparison of various cloud simulators available in the cloud computing,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, pp. 347–350, 2013.

[28] L. Wallentin, M. Happenhofer, C. Egger, and J. Fabini, “Xml meets simulation: concepts and architecture of the ibksim network simulator,” *SNE*, vol. 41, 2010.